

**CHRIS CRAWFORD ON INTERACTIVE
STORYTELLING**

ITALIAN SUMMARY
Anselmo Zoccali

PART I – FROM STORY TO INTERACTIVE STORYTELLING

STORY

Il primo capitolo del libro inizia con una analisi del concetto di *Storytelling* (termine che non ha un corrispettivo in lingua italiana, e che sta ad indicare la predisposizione naturale dell'uomo a comunicare attraverso storie e racconti) e di come esso va a collocarsi all'interno degli schemi mentali individuati nel cervello umano. Crawford ritiene che lo *Storytelling* sia un impulso nato dall'interazione di due delle macroaree in cui viene suddivisa l'attività cerebrale, ovvero quella **sociale** (che gestisce l'ambito relazionale di un individuo) e quella del **linguaggio** (che permette la comunicazione, attraverso il processo sequenziale della parola).

Ma come mai l'impulso dello *storytelling* è diventato un elemento fondamentale di quasi tutte le culture presenti e passate?

Crawford risponde partendo da una domanda: pensiamo all'organizzazione delle informazioni all'interno del cervello umano come ad un insieme di nodi (i concetti) legati in diversi modi l'uno all'altro, che vada a formare una sorta di "pattern" di idee. Il linguaggio è invece una forma di comunicazione, come già accennato, di tipo sequenziale. Come è possibile, quindi, passare informazioni ad uno schema mentale strutturato "a pattern" attraverso una forma di comunicazione sequenziale?

E' qui che entra in gioco il "racconto": il racconto infatti non è un concetto isolato, inserito in mezzo ad un pattern già formato di idee e concetti; esso è, al contrario, un sistema di concetti e fatti connessi tra loro, che quindi ben si adatta all'organizzazione delle informazioni all'interno del cervello umano.

Una consistente parte del capitolo è invece dedicata invece al concetto di racconto: Crawford articola questa parte del capitolo in "lezioni", ognuna delle quali termina con una breve summa dell'argomento affrontato. Non si tratta di regole precise, ma piuttosto di "tracce" e di suggerimenti utili a circoscrivere con sufficiente precisione il concetto del termine:

- i racconti sono strutture complesse che devono soddisfare molti requisiti difficili da razionalizzare. Sta di fatto che non tutte le sequenze ordinate di avvenimenti possono essere considerate un racconto;
- i racconti parlano della cosa più affascinante che esista nell'universo: le persone;
- i *puzzles* (intesi ovviamente anche in senso figurato) non sono una componente imprescindibile di un racconto;
- lo spettacolo non fa un racconto (Crawford si riferisce alla tendenza, ormai dominante nell'ambito del cinema hollywoodiano, a considerare la spettacolarità degli effetti speciali come il cardine su cui costruire una trama);
- il pensiero visuale non deve dominare il racconto: esso deve essere integrato con le conoscenze che ognuno di noi ha assimilato nel corso della sua vita (ad esempio, se uno spettatore vede un albero su uno schermo, integrerà ciò che vede con le sue conoscenze in campo botanico, per avere un quadro informativo il più possibile completo ed esauriente);
- il pensiero spaziale non è importante nel racconto: l'ambientazione di un racconto non deve essere pensata come una mappa con coordinate spaziali precise, ma come una serie di ambientazioni, o *stage*, completamente slegate tra loro. Ogni ambientazione va considerata come un universo a sé stante, privo di qualunque rapporto di spazialità con gli altri.

INTERACTIVITY

Dopo aver introdotto il concetto di racconto, viene introdotto il concetto di interattività, da Crawford definita *un processo ciclico tra due o più agenti attivi, durante il quale ogni agente alterna le seguenti attività: ascoltare, pensare e parlare*. Ovviamente questi tre termini vanno interpretati in senso lato: è ovvio che un computer non potrà mai ascoltare, né tanto meno pensare. Ma esso svolge comunque delle attività che possono essere messe in relazione a quelle usate da Crawford nella definizione: “accettare dati” (ascoltare), “processare dati” (pensare), “restituire il risultato” (parlare).

Crawford ci tiene a precisare che, al fine di progettare una buona interazione, non bisogna tralasciare nessuno di questi tre aspetti: una interazione risulta credibile solo quando ognuno di essi viene svolto da tutti gli agenti coinvolti. Questo inoltre circoscrive efficacemente l’ambito delle attività definibili come “interattive”: la visione di un film non potrà mai essere considerata una interazione, in quanto un film può essere un buon “parlatore” (suscitando emozioni attraverso le immagini o i dialoghi), ma non saprà mai un ascoltatore delle emozioni dello spettatore.

La peculiarità che un “artista interattivo” deve assolutamente possedere secondo Crawford, è quella da lui stesso definita *second-person insight*: ogni artista, ovviamente, è consapevole che la sua opera verrà giudicata da un pubblico, ma è anche vero che durante la creazione dell’opera l’artista prende decisioni secondo quello che è il suo gusto, mettendo in gioco il giudizio della gente solo in un secondo momento. Molto spesso, inoltre, lo scarso successo di un’opera non va a ledere l’integrità di un’artista, anzi, in casi limite essa ne esce addirittura rafforzata. L’artista interattivo, invece, non può permettersi il lusso di anteporre la sua visione del mondo a quella di chi usufruirà del suo lavoro: egli, al contrario, deve fare propria la visione degli utenti, entrare nelle loro menti e capire cosa essi si aspettano dalla sua opera e, nei limiti del possibile, cercare di accontentarli. Questa abilità è appunto la *second-person insight*.

Chi vuole arrivare a possederla deve prima di tutto essere consapevole dei meccanismi di funzionamento della mente umana: ogni artista interattivo deve avere chiaro in mente il concetto di *memoria associativa*: le informazioni nella mente umana non sono “compartimenti stagni” completamente indipendenti, ma al contrario sono organizzate per associazione di idee; ogni concetto sarà inevitabilmente collegato ad altri concetti (ad es. cane – animale). Bisogna però considerare che questa sorta di “rete concettuale” non è la stessa per ogni essere umano, anzi, la sua composizione dipende da un numero enorme di fattori (età, sesso, luogo di residenza, esperienze, ecc.); il difficile compito dell’artista interattivo è quello di individuare tutte le possibili discrepanze tra individuo ed individuo e cercare di eliminarle attraverso l’elaborazione di “denominatori comuni” concettuali.

Il capitolo si chiude con una riflessione sui *gradi di interattività*: essa infatti non va considerata un concetto binario (acceso – spento, 0 – 1), ma un concetto aritmetico (come, ad esempio, il peso corporeo). Una applicazione può essere più o meno interattiva di un’altra; i fattori in gioco sono tre, ovvero:

- **velocità**: per fare sì che l’utente rimanga costantemente dentro la comunicazione bisogna dargli delle reazioni in tempo reale; tempi lunghi di reazione da parte dell’applicazione comportano una perdita di attenzione da parte dell’utente, il quale entrerà ed uscirà continuamente dall’interazione, facendole perdere credibilità ed efficacia;
- **profondità**: col termine profondità si intende tutte le operazioni che esulano dalla pura razionalità ed entrano nell’ambito delle dinamiche di relazione tra individui. Una adeguata simulazione di queste ultime aggiunge grande credibilità all’interazione;
- **scelta**: la scelta viene identificata da Crawford come il momento decisivo dell’attività intellettuale: ogni nostro pensiero viene poi concretizzato nella scelta di agire in un modo piuttosto che in un altro. E’ dunque essenziale, per una buona interazione, che all’utente venga fornito un numero adeguato di scelte; non esiste un numero assoluto, il

criterio che l'artista deve utilizzare è quello di fornire all'utente tante scelte quante sono quelle da lui immaginabili. Se ad ogni suo pensiero viene fatta corrispondere una effettiva possibilità di azione, egli avrà l'impressione di potersi muovere in totale libertà all'interno dell'interazione.

INTERACTIVE STORYTELLING

Una volta introdotti i due concetti di *storytelling* e di interattività, Crawford passa ad affrontare uno degli argomenti chiave del libro, ovvero l'**interactive storytelling**: è davvero possibile far coesistere all'interno di una stessa opera questi due concetti, apparentemente inconciliabili?

Innanzitutto bisogna confrontarsi con pareri autorevoli che rifiutano ogni rapporto tra interattività e narrativa. Effettivamente, se si pensa all'enorme talento ed energia creativi necessari alla costruzione di un *plot* avvincente, un'intrusione da parte del lettore nel determinare le scelte dei personaggi potrebbe essere deleteria per la buona riuscita del racconto. A questo ostacolo va aggiunta l'attitudine notoriamente "*anti-plot*" di molti artisti interattivi, soprattutto nell'ambito dei giochi per computer (si pensi ad un gioco famoso come *Doom*, dalla trama praticamente inesistente). Questa dicotomia ha radici molto antiche, e può essere accostata al dibattito teologico riguardante il contrasto tra libero arbitrio (in questo caso l'interattività) ed il determinismo (ovvero il *plot*): si tratta di concetti diametralmente opposti, che non potranno mai coesistere nello stesso contesto.

Date queste premesse, sembra sparire ogni possibilità di relazione tra questi due concetti, eppure anche le applicazioni interattive possono essere considerate, in un certo qual modo, deterministiche: l'unica differenza è che invece degli eventi, ad essere determinate sono le regole che i personaggi dovranno seguire nel corso del racconto. Se dunque è vero che interattività e *plot* sono concetti incompatibili, ad un maggiore livello di astrazione possiamo invece trovare dei punti di contatto. La narrazione interattiva non segue un *plot*, ma piuttosto un *metaplot*, il quale guida le azioni dei personaggi non determinandole direttamente, ma circoscrivendole ad una serie di regole e possibilità finite.

Per concludere la questione sollevata nel paragrafo precedente, il compito dell'artista interattivo è quello di mettere l'utente davanti ad un certo numero di scelte nel corso della narrazione, ognuna delle quali deve essere in grado di mantenere alto l'interesse per il racconto.

Vi sono tre problemi che automaticamente vengono sollevati da questa affermazione:

- **Come si fa a generare un numero sufficiente di decisioni "interessanti"?**

Innanzitutto bisogna evitare le decisioni ovvie: per utilizzare un esempio, se in una ipotetica narrazione la regina venisse rapita, sarebbe inutile porre il re (ovvero l'utente) di fronte ad una scelta come questa:

scelta A: parti alla ricerca della tua amata regina per salvarla;

scelta B: non fare niente e resta nel tuo castello.

Chiunque sia un minimo interessato a continuare la sua avventura interattiva sceglierà senza indugiare la prima strada. L'intelletto dell'utente, invece, deve sempre essere messo alla prova ad ogni bivio che gli si pone davanti; solo così il suo livello di attenzione può mantenersi costante.

- **Come si fa ad evitare di inserire decisioni "noiose"?**

Al fine di evitare di porre l'utente di fronte a scelte noiose e poco rilevanti per il proseguimento della narrazione, bisogna imparare a riconoscere quali sono i "nodi drammatici", quelli che sono in grado di dare una vera svolta al racconto, concentrandosi esclusivamente su di essi. Ad esempio, se il protagonista ha già deciso di arrendersi ad

un'imboscata nemica, non sarà necessario chiedergli se vuole gettare l'arma a terra oppure no; sarà una naturale conseguenza dell'essersi arreso.

Ovviamente, se si vuole mantenere alto il tasso di coinvolgimento emotivo dell'utente, bisognerà strutturare la storia in modo da non far passare mai troppo tempo tra una decisione e la successiva, "compattando" il racconto in presenza di lunghe parti "non interattive".

- **Come si fa a creare uno *storyworld* interessante?**

Cominciamo col dare una definizione al termine *storyworld*: esso sta ad indicare il mondo in cui si svolge la narrazione, con i suoi personaggi, le sue regole e tutte le sue potenziali *storylines* (con tale termine si intende una serie concatenata di eventi che si svolgono in tale mondo, e che vanno a formare un *plot*). Ed è proprio il concetto di *storyline* che bisogna dimenticare, se si vuole costruire uno *storyworld* avvincente: uno *storyworld* progettato su una singola o su poche *storylines* risulterà irrimediabilmente limitato e noioso; al contrario esso deve essere un vasto universo di possibilità, di molteplici e differenti vicende che si intrecciano ed influenzano a vicenda.

Vi sono alcune applicazioni informatiche, spesso definite come "interattive", nelle quali l'utente prende tutte le decisioni prima dell'inizio della storia stessa. Spesso il suo compito è quello di definire le caratteristiche del suo personaggio, o di compilare un questionario. In base agli input forniti, l'applicazione elabora una storia completa.

Per quanto interessanti, tali applicazioni non possono essere definite in alcun modo interattive: il loro funzionamento va infatti a cozzare con la definizione di interattività fornita da Crawford nel secondo capitolo. L'utente parla una sola volta col computer, il quale una sola volta elabora ed una sola volta risponde; manca la ciclicità di queste tre operazioni, elemento fondamentale dell'attività interattiva. La storia viene creata tutta nello stesso momento, e non attraverso un progressivo scambio di informazioni tra applicazione ed utente.

Per essere certi di trovarsi di fronte ad una applicazione interattiva, bisogna assicurarsi che tale applicazione assembli la storia partendo da piccoli frammenti di racconto, da Crawford definiti *substories*. La lunghezza di questi frammenti non è importante, la cosa fondamentale è che ogni *substory* abbia al suo interno una componente drammatica, una scelta che l'utente deve essere costretto a fare al fine di proseguire nel racconto.

PART II – STYLES OF THINKING

ABSTRACTION

Come già accennato nella prima parte del libro, l'astrazione è un concetto fondamentale nell'ambito dell'*interactive storytelling*. Il *plot* viene rimpiazzato da una rete di diverse possibilità di sviluppo narrativo, le quali però, alla fine del racconto, servono a veicolare tutto lo stesso messaggio.

Non bisogna infatti confondere il *plot* (ovvero la concatenazione di eventi che va a formare una *storyline*) con il messaggio che un racconto vuole veicolare: il film *Star Wars*, ad esempio, parla della crescita interiore di un individuo, il quale, affrontando pericoli e difficoltà, riesce a superare le sue paure e a diventare finalmente un uomo. Un racconto, se portato ad un alto livello di astrazione, trova la sua ragion d'essere nelle verità che, attraverso una storia fittizia, riesce a trasmettere allo spettatore; il *plot* non è altro che uno dei tanti possibili istanziamenti di queste verità.

L'*interactive storytelling* si basa proprio su questa concezione astratta di racconto, visto non come una serie concatenata di eventi, ma come un messaggio che necessita di essere istanziato per poter essere inviato alla mente dello spettatore (il quale dovrà fare lo sforzo di astrarre il contenuto dell'istanza e da esso risalire alla verità originaria che la storia vuole comunicare).

Ciò che differenzia il racconto interattivo da questo concetto di racconto è l'utilizzo del computer: spetta a lui infatti il compito di "istanziatore" del messaggio: rendere un computer un buon istanziatore richiede un notevole sforzo di creazione algoritmica, la quale corrisponde ad un livello di astrazione, nella gestione dello *storyworld*, ancora più elevato. Inoltre non bisogna dimenticare che spesso le applicazioni di narrazione interattiva tengono conto dei bisogni e degli interessi dell'utente, aumentando dunque il suo coinvolgimento emotivo.

VERB THINKING

Il nostro pensiero si articola in due tipologie, le quali si basano sui due elementi fondamentali di ogni lingua parlata esistente: i nomi (*noun thinking*) ed i verbi (*verb thinking*). Ogni concetto da noi pensato può essere concepito in entrambi i modi. Pensiamo ad un computer: secondo la tipologia di pensiero *noun thinking* esso è un insieme di dati. La tipologia di pensiero *verb thinking* darà invece più importanza a quello che il computer fa, e dunque alla sua capacità di processare tali dati.

La nostra mente è istintivamente orientata verso il *noun thinking*, nonostante la tipologia di pensiero più efficace sia sicuramente la *verb thinking*: essa infatti, non essendo legata ai nomi, ma bensì alle azioni, permette un livello di astrazione infinitamente superiore. L'*interactive storytelling* non è concepibile senza una modalità di pensiero incentrata sulle azioni: esso si basa sulle scelte, e l'oggetto di una scelta non è un nome, ma bensì un'azione (una persona sceglie di *fare* qualcosa). Ancora una volta l'astrazione si riconferma un concetto imprescindibile nell'ambito della narrazione interattiva.

Applicare il *Verb thinking* all'*interactive storytelling* significa dunque avere una percezione delle cose basata non su quello che sono, ma su quello che fanno: in tal modo si smette di pensare "a compartimenti stagni" e si comincia ad intuire che ogni cosa esistente nell'universo è connessa alle altre attraverso una intricata rete di relazioni causa/effetto.

Vi è un solo linguaggio attraverso il quale tali relazioni possono essere comprese ed espresse con chiarezza: il linguaggio matematico. Esso infatti è l'unico che può garantire un livello di astrazione

tale da permettere l'utilizzo di modelli adattabili alle infinite combinazioni e situazioni che possono verificarsi all'interno di un racconto. Attraverso la definizione di variabili associabili a concetti e con l'utilizzo di operatori matematici, è possibile rappresentare (ovviamente con una certa approssimazione) ogni tipo di situazione, oggetto, sentimento, qualunque cosa appartenga all'universo narrativo.

Supponiamo, ad esempio, di voler rappresentare l'"eroismo" in termini matematici. Una formula appropriata potrebbe essere la seguente:

$$\text{Eroismo} = \text{Coraggio} + \text{Altruismo}$$

La variabile `Eroismo` sarà dunque la somma dei due fattori; una volta definita, potrà essere usata in una formula come la seguente:

```
IF (Eroismo > 50) THEN
    FaiLaCosaGiusta
ELSE
    NonFareNiente
```

Ovviamente gli operatori matematici devono essere usati con pertinenza: l'uso della addizione (o sottrazione) è opportuno solo quando i fattori in gioco possono compensarsi tra loro (svolge la funzione dell'operatore logico OR). E' quello che accade, ad esempio, nell'esempio della variabile `Eroismo`. Con la moltiplicazione (o divisione) la questione è più complessa: essa va utilizzata solo quando entrambi i fattori che concorrono a determinare il valore di una variabile rivestono un ruolo fondamentale nel processo. Consideriamo, ad esempio, la variabile `Vendetta` come il risultato di una moltiplicazione tra i fattori `Rabbia` e `Malvolenza`: essa non potrebbe esistere senza la l'apporto di entrambi. Una persona arrabbiata, ma che non ha motivo di provare rancore verso nessuno, non architetterà mai un piano di vendetta; allo stesso modo agirà una persona che prova rancore per un'altra persona, ma completamente serena.

Inoltre gli operatori matematici possono essere usati per sofisticare ulteriormente le formule, in modo da renderle più credibili. Supponiamo che il nostro protagonista abbia vissuto brutte esperienze che lo rendono titubante ad intraprendere nuove avventure: questo fattore gioca a sfavore della variabile `Eroismo`; supponendo di poter assegnare un valore numerico a queste brutte esperienze (20), una formula più accurata potrebbe essere la seguente:

$$\text{Eroismo} = \text{Coraggio} + \text{Altruismo} - 20$$

Questi fattori esterni vengono chiamati *weighting factors*. Essi si dividono in tre tipologie:

- **additive** (operatori + e -);
- **multiplicative** (operatori * e \);
- **exponential**: per capire il funzionamento di questi ultimi è necessario ricorrere ad un esempio: supponiamo di dover definire una variabile chiamata semplicemente `var`, attraverso la moltiplicazione di due fattori `fac1` e `fac2` e con l'utilizzo di un **multiplicative weighting factor**:

$$\text{var} = 2 * \text{fac1} * \text{fac2}$$

Sorge un problema: supponiamo che un fattore valga x e l'altro y . Pur invertendo il valore dei fattori il risultato non cambierebbe, in quanto $2 * x * y = 2 * y * x$. Com'è dunque possibile, in questo caso, mettere in relazione il peso di un fattore col peso dell'altro? Per

risolvere questo problema si utilizzano le potenze. Esse possono essere assegnate ad un singolo operatore, ottenendo così risultati diversi a seconda del valore di quest'ultimo:
 $x^2 * y \neq y^2 * x$.

Ovviamente l'idea di ridurre la natura e la psiche umana ad un insieme di formule matematiche potrebbe suscitare non poche perplessità. Ma bisogna ricordarsi che si sta parlando di *fiction* e non di realtà. Certo, l'uomo è un essere estremamente complesso, ma i personaggi che possiamo ammirare nei film o di cui possiamo leggere le gesta in un libro sono quasi sempre semplificazioni di quella che è l'intricatissima natura umana. Spesso, anzi, essi non sono altro che l'istanza di una determinata caratteristica (cattiveria, bontà, stupidità, ecc.).

PART III – STRATEGIES FOR INTERACTIVE STORYTELLING

SIMPLE STRATEGIES THAT DON'T WORK

La terza parte, incentrata sulle strategie per la creazione di applicazioni di narrazione interattiva, comincia con un elenco di soluzioni, progettate dal 1980 ad oggi, ritenute da Crawford fallimentari. La prima strategia è quella definita **branching tree**, ovvero la classica ramificazione ad albero: ogni nodo narrativo può svilupparsi in due diverse direzioni, le quali a loro volta presenteranno una ulteriore ramificazione, e così via. La critica mossa da Crawford a questo sistema riguarda soprattutto il rapporto tra carico di lavoro (e di memoria occupata) ed il risultato, tutt'altro che conveniente: per progettare una narrazione interattiva a 10 nodi occorrerebbe già una quantità di lavoro notevole, che in ogni caso lascerebbe insoddisfatto l'utente (10 passaggi sono un numero decisamente limitato). Un modo per rendere questa strategia più interessante è, come sempre, ricorrere all'astrazione, ovvero considerare i nodi narrativi non come un evento particolare, ma come situazioni standard che possono ricorrere più volte in una narrazione (ad esempio un bacio in un racconto romantico); lo schema potrebbe così diventare una struttura modulare da utilizzare più volte nella stessa trama narrativa, ogni volta in situazioni diverse, a seconda di come si è evoluta la storia.

Il secondo metodo è il **foldback scheme**: il nodo iniziale si divide in due strade differenti, i quali però si ricongiungono successivamente in un unico nodo. Secondo Crawford questa soluzione non è altro che una truffa, una presa in giro ai danni dell'utente. Vi è solo l'illusione dell'interattività, ma alla fine si è obbligati a percorrere l'unica strada possibile per arrivare alla fine del racconto. L'unico modo per rendere il **foldback scheme** un metodo davvero interattivo (sempre comunque poco efficace ed in fin dei conti fallimentare) è fonderlo con la struttura ad albero analizzata in precedenza, in modo da rendere le varie ramificazioni più varie ed interessanti.

Tra le altre strategie elencate da Crawford vi sono inoltre la **constipated story** (narrazioni che per proseguire richiedono il superamento di una prova da parte dell'utente) e gli **storified games** (videogiochi che si sviluppano seguendo un ipotetico filo narrativo), anch'essi metodi che mirano a simulare una interattività che in realtà non c'è, in quanto non si presenta mai la possibilità, per l'utente, di scegliere come proseguire all'interno della narrazione: tutto quello che può fare è superare delle prove prestabilite che gli consentano di passare al prossimo *step* narrativo.

ENVIRONMENTAL STRATEGIES

Una strategia interessante è quella definita da Crawford **Environmental Strategy**: secondo una teoria elaborata da alcuni *game designers*, la creazione di un sistema sufficientemente complesso potrebbe portare lo stesso sistema, nel tempo, a sviluppare fenomeni e comportamenti che i creatori non si sarebbero mai aspettati durante la progettazione. Applicando questa teoria all'ambito dell'*interactive storytelling*, c'è chi sostiene che da alcuni di questi sistemi potrebbero addirittura emergere spontaneamente dei racconti o si potrebbe comunque sviluppare un "terreno fertile" per la creazione di un sistema narrativo interattivo (**emergent story**).

Crawford mantiene comunque una posizione molto scettica riguardo ad una tale eventualità, e ci mette in guardia: "*emergence is not the same thing as magic*". Ma allo stesso tempo ammette che, prestando la dovuta attenzione a diversi parametri (composizione dello *storyworld*, natura degli agenti, la struttura del racconto), potrebbero crearsi le basi per l'esistenza di un sistema narrativo "spontaneo". Ovviamente questo non fa della *environmental strategy* un metodo valido: non si può certo ritenere affidabile una strategia che basa il suo funzionamento o meno sulla pura e semplice casualità.

DATA-DRIVEN STRATEGIES

Le *data-driven strategies* consistono nello scomporre la narrazione in unità elementari, le quali verranno in seguito combinate assieme da specifici *engines*, in base agli input forniti dall'utente. I dati che provvedono a fornire agli *engines* le informazioni necessarie per un assemblamento pertinente delle unità vengono chiamati *connectivity data*: la loro creazione è tutt'altro che semplice, in quanto richiede, in prima istanza, una progettazione completa e dettagliata dell'ontologia dello *storyworld* che si intende utilizzare per l'applicazione interattiva in questione. Devono essere messe in relazione, infatti, tutte le singole unità presenti al suo interno; un'operazione che richiede un enorme sforzo ed un grande carico di lavoro.

I tentativi di creazione di sistemi di connettività (*connectivity systems*) efficaci sono stati diversi, ma l'unico ad essere stato effettivamente utilizzato in una applicazione pratica è stato quello elaborato dal linguista **Vladimir Propp**: l'oggetto principale dei suoi studi sono state le favole provenienti dalla cultura popolare russa; analizzandole, Propp ha notato come esse presentino numerosi elementi (personaggi, situazioni, ambientazioni) in comune, da lui definiti *funzioni universali* (egli ha in seguito stilato un elenco di queste ultime, contenuto nella sua pubblicazione *La morfologia della favola*). Sperimentando ogni possibile combinazione tra tali funzioni, Propp è riuscito a ricostruire ognuna delle 100 favole che avevano costituito il suo campione di analisi. Nonostante Crawford ritenga questo sistema eccessivamente semplicistico, esso è stato utilizzato in una applicazione progettata da due ricercatori del Trinity College di Dublino, con risultati, ammette lo stesso Crawford, sorprendenti.

Il sistema di connettività di Propp non è comunque l'unico a basarsi sulla cosiddetta "logica del database": nel libro vengono ricordati anche i *cataloghi di Arne-Thompson* ed il sistema elaborato da **Georges Polti**. Ma essi, pur basandosi su logiche di funzionamento molto simili a quella elaborata da Propp, non hanno mai trovato riscontro in nessuna applicazione pratica.

LANGUAGE-BASED STRATEGIES

Il procedimento maggiormente utilizzato per la creazione di un'applicazione di *interactive storytelling* consiste nel modellare lo *storyworld* e le leggi che regolano l'andamento della narrazione, e successivamente convertire il risultato in un linguaggio comprensibile dall'utente. Questo passaggio, agli occhi di Crawford, è inutile: perché non fare in modo che sia il linguaggio stesso il nucleo (*core*) dell'intero processo?

Il problema principale nell'applicare una strategia basata sul linguaggio risiede nel fatto che il linguaggio naturale di ogni individuo riflette l'enorme complessità della nostra percezione della realtà: ogni termine, ogni frase da noi detta va relazionata al contesto, al mondo che ci circonda, bisogna tenere conto delle ambiguità di senso di alcuni termini, le espressioni figurate, e così via. La mole di conoscenza da caricare su un ipotetico database sarebbe ingestibile per qualunque processore.

La soluzione a questo problema è solo una: bisogna creare dei "sottolinguaggi" (*sublanguages*) che presentino un vocabolario di termini comprensibili sia dall'uomo che dal computer. Si tratterà dunque di un vocabolario molto ristretto (si può arrivare addirittura a meno di 2000 termini per ottenere un sottolinguaggio soddisfacente). Ovviamente vi sono diverse strategie con le quali è possibile rendere un vocabolario più efficace: basti pensare ad esempio alla quantità di prefissi e suffissi che è possibile applicare ai vocaboli:

- prestabilire
- fratellanza
- sincerità

- sfortunato

L'aggiunta di un piccolo set di tali estensioni, unita alla progettazione di un algoritmo in grado di gestirle in relazione con i termini del vocabolario, potrebbe giovare notevolmente all'efficacia dell'applicazione.

Un altro problema relativo alla creazione di *sublanguages* è quello della grammatica: ogni linguaggio naturale esistente al mondo possiede delle regole sintattiche tremendamente complicate; basti pensare ai verbi nella lingua inglese, molti dei quali irregolari. La soluzione, in questo caso, consiste nel ricorrere ad una grammatica semplificata, la quale risulterebbe sicuramente scorretta ed inappropriata nell'ambito della comunicazione in linguaggio naturale, ma che può rivelarsi molto utile nella comunicazione programmatore – applicazione: si tratta di una sorta di *creolo*, il quale non presenta coniugazioni o declinazioni. Qui di seguito è riportato un esempio:

Linguaggio naturale: "I went to the store" **Sottolinguaggio:** "I been go store"

La chiarezza del messaggio, nel passaggio dal linguaggio naturale al sottolinguaggio, è rimasta invariata., e ciò è sufficiente per considerare questa soluzione efficace.

L'introduzione dei *sublanguages* risolve gran parte dei problemi di *misunderstanding* che possono incorrere nel corso dell'interazione uomo – macchina, ma rimane pur sempre un certo margine di rischio. L'ultimo *step* verso la totale affidabilità è costituito dall'***inverse parsing***: tale metodo consiste nel consentire all'utente di comporre la frase un pezzo alla volta, suddividendola in unità elementari; ad ogni *step*, sarà il computer a fornire all'utente il menu delle possibilità con le quali continuare la frase. In questo modo si avrà la sicurezza che il messaggio finale inviato dall'utente alla macchina non potrà essere misinterpretato dall'applicazione, in quanto sarà essa stessa ad aver guidato l'utente nella formulazione della frase.

PART IV – CORE TECHNOLOGIES FOR INTERACTIVE STORYTELLING

PERSONALITY MODELS

Con il termine *personality model* si intende una struttura dati contenente tutte le informazioni necessarie a definire un personaggio. Non bisogna commettere l'errore di confonderlo con la nozione di *personality modeling* utilizzata in psicologia: ci sono stati diversi tentativi di applicazione di metodi psicologici all'*interactive storytelling*, ma i risultati si sono sempre rivelati scadenti. Questo perché la personalità delle persone "reali" è molto diversa da quella dei personaggi che solitamente popolano un racconto: essi prendono decisioni audaci e spesso incoscienti, si spingono sempre al limite delle loro capacità e scelgono sempre l'azione piuttosto che la tranquillità; tutte caratteristiche molto rare negli individui che incontriamo e con cui interagiamo nella vita di tutti i giorni.

Nei fattori utilizzati per definire un *personality model* devono coesistere le seguenti caratteristiche: la completezza; un *personality model* deve tenere conto di tutti i potenziali comportamenti che lo *storybuilder* vuole inserire all'interno dello *storyworld*; deve essere sicuro che i personaggi possano agire ed interagire in modo completo e pertinente in ogni situazione possibile all'interno dell'universo creato per loro.

La concisione, diretta conseguenza della completezza; è infatti inutile inserire, all'interno di un *personality model*, comportamenti che non potranno mai essere applicati in un determinato *storyworld*: nel caso, ad esempio, di uno *storyworld* per bambini, sarà inutile includere al suo interno qualunque riferimento sessuale, in quanto esso non verrà mai utilizzato in quel determinato universo narrativo.

Inoltre le variabili in un *personality model* non devono mai sovrapporsi, altrimenti sarà difficile, in molte situazioni, scegliere quale variabile applicare: prendiamo in considerazione le variabili *DiBuonUmore* e *Amichevole*; questi due fattori non sono certo concettualmente identici, ma molto spesso tendono a sovrapporsi, e soprattutto è difficile pensare a delle situazioni in cui sia preso in considerazione uno solo di essi. Distinguerli risulta dunque inutile ai fini dell'efficienza del modello, e non fa altro che renderlo inutilmente più pesante.

Esistono cinque macrocategorie di variabili che andrebbero utilizzate in ogni *personality model*:

- ***intrinsic***: tale categoria include tutte le caratteristiche personali di ogni personaggio (ad esempio avarizia, orgoglio, forza, ecc.);
- ***mood***: include le variabili riferite agli stati emozionali a cui un personaggio può essere soggetto (rabbia, gioia, ecc.);
- ***volatility***: le *volatility variables* governano la "prontezza" e facilità con cui possono variare le *mood variables* di un personaggio;
- ***accordance***: le *accordance variables* governano il modo in cui possono cambiare le relazioni tra i vari personaggi;
- ***relationship***: questo tipo di variabili include le relazioni che ogni personaggio instaura con gli altri personaggi dello *storyworld*.

Le variabili delle prime quattro categorie vengono definite ***first-person variables***, in quanto esse vengono spesso rappresentate come voci inserite in un array mono-dimensionale: *Avarizia[Personaggio]*; le variabili della categoria *relationship* invece vengono definite ***second-person variables***, in quanto rappresentate da un array bi-dimensionale: *Affetto[Personaggio, AltroPersonaggio]*. Questo perché sono variabili che non riguardano solo il singolo personaggio, ma anche i personaggi con cui quest'ultimo viene messo in relazione.

Uno dei più ostici problemi nell'ambito del *personality modeling* riguarda la polarità delle variabili: le variabili unipolari sono quelle il cui range di valori può variare da 0 ad un numero positivo qualsiasi; le variabili bipolari sono invece quelle il cui valore può variare da un minimo negativo ad un massimo positivo (ad esempio 1 e -1).

Non c'è una tipologia di variabili più appropriata dell'altra (anche se Crawford predilige l'utilizzo di variabili bipolari), tutto dipende dalle modalità di progettazione utilizzate dallo *storybuilder* e dai fattori in gioco all'interno dello *storyworld*; inoltre nulla vieta di utilizzare entrambe le tipologie di variabili all'intero della stessa applicazione.

Ogni reazione di un personaggio ad un evento che avviene all'interno dello *storyworld* deve includere due elementi: una scelta tra le possibili azioni che il personaggio può intraprendere ed una alterazione dei valori delle variabili relative a quel personaggio. Il secondo elemento potrebbe sembrare, a prima vista, estremamente difficile da determinare; vi sono però delle procedure che possono calcolare automaticamente questa variazione di valori, basate su modelli matematici.

DRAMA MANAGERS

Il raccontare storie richiede senza ombra di dubbio un narratore: l'*interactive storytelling*, dunque, presume l'esistenza di una sorta di "algoritmo narratore" all'interno del software, definito *drama manager*.

Il primo compito del *drama manager* consiste nel monitorare il progresso della storia. Il modo più semplice di agire in questo frangente è utilizzare le cosiddette *overview variables*, ovvero valori che l'*engine* calcola al fine di valutare lo stato complessivo dello *storyworld*: esso ad esempio potrebbe sommare tutte le variabili *Affetto* dei personaggi per calcolare il "livello di affetto" complessivo dello *storyworld* in quel determinato momento.

In base ai risultati ottenuti da questo primo *step*, il *drama manager* dovrà decidere se e come intervenire nello sviluppo degli eventi: l'*engine* potrebbe rilevare un certo numero di azioni inutilizzate all'interno dello *storyworld* e spingere l'utente verso quelle "strade inesplorate", oppure basarsi su linee guida (*dramatic templates*) inserite nella progettazione dell'applicazione, o ancora basarsi sulle *data-driven strategies* analizzate in precedenza.

Il passo finale è mettere in atto le decisioni prese ed intervenire direttamente sullo *storyworld*. Vi sono diversi metodi attraverso i quali compiere questo ultimo *step*:

- **manipolazione ambientale**: un metodo consiste nell'intervenire direttamente sull'ambiente dello *storyworld*, causando eventi che indirizzino l'utente verso la strada voluta dallo *storybuilder*. Si tratta di una tecnica estremamente invasiva, la quale deve essere usata con attenzione: l'utente non deve mai essere costretto ad intraprendere un'azione piuttosto che un'altra, ma al massimo spinto, influenzato dall'ambiente circostante;
- **inserimento o cambio di obiettivi**: un metodo più sottile consiste nell'assegnare all'utente nuovi obiettivi, o cambiarli. Ovviamente i nuovi obiettivi devono essere compatibili con le caratteristiche del personaggio, al fine di mantenere la storia credibile;
- **cambiamento delle personalità**: la soluzione più elegante è quella di alterare le personalità degli altri personaggi presenti nello *storyworld* in modo tale da influenzare le decisioni dell'utente;
- **controllo temporale**: è anche possibile creare una sorta di "tracciato" narrativo inserendo una sequenza di eventi distribuiti nel tempo che guidino l'utente attraverso le varie fasi del racconto;
- **infrangere la quarta parete**: il metodo in assoluto peggiore e più invasivo consiste nello svelare la finzione, abbattere la fantomatica "quarta parete" ed avvisare direttamente l'utente che le sue azioni non sono funzionali al proseguimento della narrazione.

Ad una analisi superficiale potrebbe sembrare che in tal modo venga meno il libero arbitrio dell'utente, ma ovviamente tutto si basa sull'abilità dello *storybuilder* nell'influenzare le sue decisioni senza per questo forzarlo ad intraprendere una azione piuttosto che un'altra. L'utente, in ogni caso, deve sentirsi libero di agire come meglio crede, senza restrizioni di sorta.

VERBS AND EVENTS

Come già accennato nei precedenti capitoli, il *verb-based thinking* costituisce un elemento imprescindibile nella progettazione di applicazioni di *interactive storytelling*: sono infatti i verbi che consentono ai personaggi di intraprendere le azioni che fanno andare avanti il racconto. Una applicazione di narrazione interattiva nella stragrande maggioranza dei casi richiede centinaia, o addirittura migliaia di verbi; come possono quindi gli *storybuilders* essere in grado di definire un così vasto range di azioni?

La risposta risiede ancora una volta nella astrazione. Ciò che occorre è infatti un approccio più generalizzato ed incentrato sulla "customizzazione" dei verbi: essi vanno nominati e classificati secondo una matrice di caratteristiche che sia uguale per ogni verbo inserito. In tal modo la gestione e l'inserimento nello *storyworld* di un considerevole numero di possibili azioni è molto più semplice e veloce.

Quando un verbo, nel corso del racconto, viene eseguito, diventa parte di un **Evento**: ogni evento, nell'ambito dell'*interactive storytelling*, deve essere riconducibile ad una funzione, ad esempio:

Maria ha dormito.

Marco ha insultato Luca.

Nel primo esempio "Maria" è considerata il soggetto dell'espressione, "ha dormito" il verbo; nella seconda, oltre a questi due parametri, si aggiunge "Luca", ovvero l'oggetto diretto, il personaggio a cui l'azione è rivolta.

Ovviamente si tratta di costruzioni elementari, le quali non possono neanche lontanamente coprire l'intera gamma di costrutti ed espressioni di una qualunque lingua parlata. Lo sforzo che deve fare lo *storybuilder* è dunque quello, nei limiti del possibile, di ridurre ogni evento dello *storyworld* ad una espressione simile.

Ma per quanti sforzi si possano fare, rimangono sempre dei costrutti particolarmente complessi che non possono essere ridotti ad una espressione come quelle riportate nell'esempio, e che non possono essere ignorati. Crawford, dopo aver escluso l'opzione delle strutture ricorsive (considerate troppo caotiche in un contesto simile), introduce come possibile soluzione il concetto di **flat data structure**: tale struttura consente di aggiungere all'espressione un certo numero di argomenti che possono essere utilizzati come oggetti indiretti, come nell'esempio seguente:

Soggetto dice (a) OggDiretto (che) OggIndiretto1 odia OggIndiretto2

Usando questa struttura è possibile ridurre ogni frase detta in linguaggio naturale ad una espressione riconoscibile da un *engine* informatico.

HISTORYBOOK AND GOSSIP

Spesso in una narrazione interattiva l'utente e gli altri personaggi, muovendosi attraverso lo *storyworld*, generano Eventi che possono influenzare le future decisioni. Per questo è necessario tenere traccia di ogni singolo Evento avvenuto nel corso della narrazione. Crawford chiama questo stoccaggio di dati *HistoryBook*: nella sua forma più semplice, esso non è altro che un elenco in ordine cronologico di tutti gli Eventi che hanno avuto luogo nello *storyworld*; ogni Evento è collegato all'Evento che l'ha generato e all'Evento che esso stesso ha generato a sua volta. Tali informazioni sono spesso utili per evitare comportamenti ripetitivi da parte dei personaggi. Non bisogna però commettere l'errore di considerare ogni *HistoryBook* una semplice sequenza lineare di avvenimenti: spesso un Evento può rivelarsi generatore di più eventi, dinamica che comporta la nascita di una più complicata struttura ad albero.

Raramente lo sviluppo drammatico di una narrazione interattiva fa riferimento solo all'esperienza diretta dell'utente all'interno del gioco; spesso molte informazioni cruciali e dall'alto coefficiente drammatico vengono comunicate indirettamente, attraverso la relazione con gli altri personaggi. Per tale motivo ogni applicazione di *interactive storytelling* che possa essere considerata efficace deve fornire un sistema di "trasmissione di informazioni" tra i personaggi della narrazione, da Crawford definito *Gossip system*.

Il punto di partenza per la progettazione *gossip system* adeguato è la creazione di un *HistoryBook*; ad ogni Evento inserito al suo interno va associato un array che contenga i nomi dei personaggi che sono al corrente di quell'evento. Ogni personaggio, a sua volta, deve tenere in memoria una lista degli eventi per lui più "invitanti" (*headline list*), che potrebbero sollecitare la sua inclinazione al gossip; saranno tali eventi ad essere da lui divulgati, se si presenterà l'occasione. I personaggi però non possono divulgare un singolo Evento o *headline*: per rendere il loro gossip interessante devono essere in grado di ricostruire tutti gli *step* che hanno portato a quel particolare avvenimento. Ciò richiede un algoritmo che sia in grado, partendo dall'*headline*, di isolare tutti gli Eventi, all'interno dell'*HistoryBook*, che hanno portato all'*headline* stesso.

Oltre a scegliere gli argomenti "caldi" da rendere oggetto di gossip, i personaggi hanno anche il compito di scegliere il loro *gossip-mate*. Questa operazione può essere facilmente automatizzata: una volta entrato in uno *Stage* ed aver preso in considerazione tutte le persone presenti al suo interno, il personaggio calcolerà, attraverso una valutazione dei *personality models* (concetto già trattato nei capitoli precedenti), chi tra loro è la persona a lui più congeniale. Dopo questa valutazione relativa, si passa ad una valutazione assoluta: il personaggio deve decidere se tale persona è a tutti gli effetti "degnata" di ricevere quella determinata informazione (valutazione che dipende non solo dal rapporto tra il personaggio ed il suo potenziale interlocutore, ma anche da caratteristiche personali del personaggio stesso, ad esempio la sua loquacità).

Ma nello sviluppo drammatico di un racconto ad essere importante non è solo l'informazione, ma anche la disinformazione, o l'informazione errata. Le bugie, ad esempio, sono molto spesso il motore drammatico di una narrazione. Implementare una bugia dal punto di vista algoritmico è molto semplice: basta inserire un valore booleano che stia ad indicare la verità o falsità di un determinato Evento; risulta invece molto più difficoltoso pianificare i casi in cui un personaggio dovrebbe mentire. Le ragioni per divulgare una bugia possono essere molteplici, e difficili da prevedere: incrinare il rapporto tra due personaggi, trarre un qualunque tipo di vantaggio o beneficio, difendersi, ecc.

Un'altra forma di *twisted information* è la rivelazione di un segreto; del resto il concetto di segreto all'interno di una narrazione interattiva ha senso solo nel momento in cui esso viene subdolamente divulgato. In questo modo è possibile accrescere la drammaticità del racconto ed indirizzarlo verso sviluppi narrativi intensi ed interessanti.

ANTICIPATION

L'implementazione del concetto di *Gossip* analizzata nel capitolo precedente non è priva di punti deboli. Supponiamo che un personaggio abbia una relazione con la moglie del suo migliore amico: applicando il processo relativo al *Gossip* illustrato in precedenza, il personaggio in questione dovrebbe immediatamente correre dal suo migliore amico ed informarlo dell'adulterio. In sostanza, pur trattandosi di un algoritmo apparentemente inattaccabile, esso può portare i personaggi a prendere decisioni quantomeno bizzarre.

Un modo per evitare questa eventualità è l'anticipazione: ogni personaggio dovrebbe essere in grado di prevedere le reazioni dell'altro, se messo al corrente di un determinato fatto. Vi sono due approcci che possono essere presi in considerazione: il primo consiste nel calcolare la probabile reazione di un personaggio nell'apprendere una certa notizia, e determinare se tale reazione sia desiderabile o no per il divulgatore. Il secondo (consigliato ed applicato da Crawford) è quello di sperimentare direttamente la reazione, senza alcun calcolo probabilistico, e determinare con sicurezza se essa sia desiderabile per colui che divulga la notizia. All'interno della narrazione ciò non avverrebbe sul piano reale, ma su un ipotetico piano virtuale, inserito all'interno dell'applicazione di *interactive storytelling*, virtuale a sua volta. Tutto ciò richiede ovviamente un enorme lavoro in termini di programmazione algoritmica, ed una buona capacità computazionale del processore; inoltre in molti casi intervengono altre infinite varianti, ciascuna delle quali richiede meccanismi di inferenza specifici.

ROLES AND SEQUENCING

Il termine "opzione", nell'ambito dell'*interactive storytelling*, sta ad indicare una possibile azione (Verbo, concetto analizzato nei precedenti capitoli) che l'applicazione offre al personaggio in reazione ad un determinato Evento. Siccome ogni Evento è caratterizzato quasi esclusivamente dal Verbo al suo interno, ogni opzione viene in realtà assegnata ad uno specifico Verbo, piuttosto che all'Evento che lo contiene.

Vi sono due modi per assegnare le opzioni ai rispettivi Verbi:

- **hard-wiring**: questa modalità consiste semplicemente nel costruire una lista di reazioni univoche (possibilmente pertinenti e plausibili) per ogni Verbo inserito all'interno dello *storyworld*;
- **soft-wiring**: la modalità *hard-wiring* potrebbe apparire inelegante, in quanto ogni assegnazione deve essere implementata manualmente. Un approccio sicuramente più sofisticato è il *soft-wiring*: esso consiste nell'assegnare ad ogni Verbo un set di tratti caratteristici che possono venire esaminati dall'*engine*, il quale deciderà, in base ai risultati ottenuti, il Verbo da associare ad ogni situazione. Si tratta però di un approccio raramente utilizzabile in applicazioni pratiche, vi sono troppi fattori da prendere in considerazione. L'approccio *hard-wiring*, allo stato attuale delle cose, risulta decisamente più pratico.

Le opzioni, come abbiamo visto, vengono assegnate ai Verbi e non ai personaggi, ma questo non significa che qualunque personaggio, in una determinata situazione, possa avere a disposizione tutte le opzioni di reazione possibili; alcune opzioni possono essere, ad esempio, strettamente legate alla natura sessuale di un personaggio, o al ruolo che egli ricopre in una determinata situazione.

Non vi sono automatizzazioni per ovviare a tale problema, l'unica soluzione è prendere in considerazione ogni singolo Evento ed i personaggi che ne prendono parte, e valutare per ciascuno di essi quali possibili opzioni sono pertinenti con il ruolo da loro ricoperto. In molti casi possono incorrere ambiguità e diverse varianti: cosa fare, ad esempio, quando la reazione di un personaggio esclude la possibilità di reazione di un altro personaggio anch'esso presente sullo *stage*? Le soluzioni sono due: l'*engine* può dare la priorità alla reazione la cui messa in atto contribuirà

maggiormente allo sviluppo drammatico della narrazione (*reaction sequencing*), oppure determinare quali reazioni possono essere evitate ed automaticamente scartarle, in modo da dover tenere in considerazione solo le reazioni inevitabili ai fini del proseguimento del racconto (*hijacking*).

La struttura più semplice per un *engine* di *interactive storytelling*, è quella ad esecuzione immediata; l'utente decide quale azione far svolgere al suo personaggio e quest'ultimo la esegue immediatamente. Ma non sempre ciò è possibile: alcune azioni, per essere messe in pratica, richiedono un certo *delay* temporale tra la decisione dell'utente ed il loro effettivo svolgimento all'interno dello *storyworld*.

La gestione di questi casi è complessa: prima di tutto l'azione, nel momento in cui il personaggio sceglie di compierla, va inserita nella *plan list*, un elenco dinamico contenente tutte le scelte prese dall'utente che non sono ancora state messe in pratica; ognuna di queste azioni ha un proprio *preparation time*, ovvero l'arco di tempo necessario affinché l'esecuzione sia possibile. Una volta esaurito tale lasso temporale, l'*engine* trasferirà l'azione nella *opportunity list*, un secondo elenco dinamico per tutte le azioni già decise dall'utente, ma le cui condizioni di attuazione non sono ancora state raggiunte; quando ciò accadrà, solo allora l'azione decisa dall'utente potrà finalmente essere attuata.

DEVELOPMENT ENVIRONMENTS

Progettare un software per l'*interactive storytelling* è un lavoro estremamente lungo e complesso; la mole di dati richiesta per la definizione di personaggi, Verbi, *Stages*, e tutte le altre componenti dello *storyworld* è enorme. Di conseguenza, gli ambienti di sviluppo (*development environments*) sono di fondamentale importanza nella progettazione di una applicazione efficace.

Molti *storybuilder* si lasciano spaventare facilmente alla vista di formule matematiche. Per questo Crawford consiglia di utilizzare interfacce che possano facilitare il lavoro degli *storybuilders*, automatizzando e rendendo sotterranei molti processi. Un esempio può essere l'eliminazione dei loop: inglobandoli all'interno di specifiche funzioni è possibile farli sparire dalla vista degli *storybuilder*, rendendo l'intero processo più intuitivo. Sono inoltre stati elaborati diversi metodi atti a semplificare i modelli matematici utilizzati nella determinazione dei parametri dei personaggi e dello *storyworld*, in modo da ridurre al minimo la presenza di operatori matematici. Non deve però mai mancare un *editing* che consenta di intervenire direttamente sul linguaggio di scripting utilizzato dall'ambiente di sviluppo; permettere agli *storybuilders* di operare a qualunque profondità nei processi computazionali è fondamentale.

E' molto importante che gli ambienti di sviluppo offrano agli *storybuilder* la possibilità di eseguire dei test in corso d'opera: una efficace *testing facility* è quella denominata *threadtest*. Un *thread* è una sequenza di Eventi che si sviluppa a partire da un determinato punto di una narrazione, ovviamente seguendo le regole dello *storyworld*. Dopo aver scelto il punto di partenza nella *storyline*, viene generato un certo numero di *threads*, i quali vengono in un secondo momento assemblati in una singola struttura statistica riguardante il probabile sviluppo della storia. Questo tipo di test è necessario per rintracciare due grandi imprevisti che possono emergere all'interno di uno *storyworld*: i *threadkillers*, ovvero nodi dai quali l'azione non può più svilupparsi in nessun modo, causando la "morte" del racconto, ed i *loopyboobies*, ovvero i classici loop in cui a volte si incagliano molte applicazioni ludiche. Una volta rintracciati questi *bugs*, lo *storybuilder* potrà eliminarli attraverso una ridefinizione del sistema causa/effetto dello *storyworld*.

Altri strumenti di analisi possono essere invece utilizzati per “ripulire” la grammatica dell’applicazione da Verbi che verrebbero scarsamente utilizzati e che dunque la appesantirebbero inutilmente. Supponiamo di avere una situazione dalla quale possono diramarsi tre differenti azioni: se durante i test una di queste tre opzioni non viene mai presa in considerazione dall’*engine*, lo *storybuilder* molto probabilmente dovrà rivalutarne l’utilità.

Inoltre molte strategie per migliorare l’usabilità delle applicazioni interattive elencate nei capitoli precedenti possono essere utilizzate anche all’interno dell’ambiente di sviluppo. Si pensi ad esempio all’*inverse parsing*: l’utilizzo di menù a cascata che guidino lo *storybuilder* nella creazione del proprio *storyworld* è ormai una caratteristica comune a qualunque *development environment* che possa vantare un minimo bacino di utenza.

PART V – APPLICATIONS

RESEARCH

Prima di esplorare nel dettaglio gli argomenti del capitolo in questione, Crawford ricorda ai lettori che gli sforzi per la ricerca nell'ambito dell'*interactive storytelling* non sono ancora mirati alla realizzazione di applicazioni pratiche. Si tratta di una disciplina molto giovane, complessa ed ancora in fase di sperimentazione; pertanto gli obiettivi di chi decide di cimentarsi in essa devono essere rivolti verso l'esplorazione di nuovi ed innovativi concetti, non verso prodotti tangibili ottenuti riciclando tecnologie già ampiamente collaudate, ma in questo caso inadatte.

Un chiaro esempio di questa tendenza al "riciclaggio tecnologico" è l'utilizzo del *boolean thinking* all'interno di molti processi legati alla narrazione interattiva: l'utilizzo di valori booleani è di indubbia efficacia nella soluzione di qualunque *engineering problem*, ma diventa estremamente limitante quando si cerca di descrivere l'ontologia di un mondo ipotetico, o di tracciare la situazione emotiva di un certo personaggio. Si tratta di situazioni che non possono essere descritte mediante il solo utilizzo di due valori (`true` o `false`).

Nonostante si tratti dunque di sforzi prevalentemente teorici, molti sono stati e sono tutt'ora i tentativi di concretizzarli, o quantomeno di inserirli all'interno di applicazioni informatiche già esistenti: già negli anni Ottanta si è cercato di applicare, attraverso il progetto *Oz*, la *agent technology* (tecnologia basata su modelli di comportamento) all'interno di determinati software. Il risultato allora più rilevante fu l'introduzione degli *help wizards*, ancora oggi parte integrante di software come Microsoft Word, seguita da molti altri esperimenti dai risultati alterni.

Altri tentativi si sono susseguiti negli anni. Di seguito ne sono riportati alcuni tra i più significativi:

- il progetto *Experience Management* ha cercato di trovare il giusto equilibrio nella, secondo molti, inevitabile dicotomia esistente tra narrazione ed interattività mediante l'utilizzo di sofisticati *foldback schemes* (strategia analizzata nei capitoli precedenti);
- il progetto *Virtual Storyteller* si basa su un sistema di agenti autonomi controllati da un *director agent* (una sorta di *drama manager* virtuale), il quale è responsabile del loro comportamento e può utilizzare tre diverse strategie per influenzarlo: disapprovandolo esplicitamente (se il *director agent* disapprova, l'agente è costretto a modificare la sua modalità d'azione), fornendo all'agente un obiettivo o modificando quello già esistente, oppure introdurre delle variazioni nell'ambiente in cui l'agente è immerso (ad esempio inserendo nuovi personaggi o oggetti all'interno dello *storyworld*);
- le *HTN-based technologies* (*Hierarchical Task Network-based technologies*) basano il loro funzionamento sull'organizzazione gerarchica degli obiettivi e dei compiti: in cima all'albero gerarchico vengono posti gli obiettivi, ognuno dei quali viene scomposto in sotto-obiettivi, i quali a loro volta vengono scomposti in singoli compiti, ovvero le unità elementari dell'azione. L'agente virtuale sarà così in grado di identificare un obiettivo e raggiungerlo attraverso un processo *step by step*. Ma la logica di funzionamento di tali tecnologie, per quanto estremamente raffinata, deriva dal *boolean thinking*, e ciò la rende inadatta a gestire determinate situazioni: essa potrebbe funzionare magnificamente per risolvere problemi meramente pratici all'interno dello *storyworld*, ma risulterebbe, ad esempio, un pessimo "gestore di emozioni".
- una delle più brillanti applicazioni pratiche di narrazione interattiva mai realizzate è sicuramente *Façade*, nata dalla collaborazione tra **Michael Mateas** e **Andrew Stern**: l'intera vicenda si svolge all'interno dell'appartamento di una coppia in piena crisi coniugale; l'utente, invitato a cena, sarà involontario spettatore di una lite furibonda tra i due, e dovrà decidere quale comportamento adottare per raggiungere il suo obiettivo (il quale può essere o tentare di riconciliare la coppia o adoperarsi per la loro definitiva

separazione). Una carta vincente di *Façade* è sicuramente l'estrema specificità della situazione, unita ad un'ambientazione estremamente essenziale e ristretta: essendoci poche possibilità d'azione all'interno dell'appartamento, è stato possibile prevederne la maggior parte in fase di progettazione. L'utente prova così una sensazione di totale libertà all'interno dello *storyworld*, in quanto ogni azione che l'ambiente può suggerirgli è stata presa in considerazione. L'illusione di realtà, inoltre, è accentuata anche dalla *real time action*, utilizzata in luogo dell'interazione "a turni".

PARENT RELATIVES

Molte applicazioni informatiche, in occasione della loro uscita sul mercato, sono state acclamate come altamente interattive pur non essendolo: è il caso, ad esempio, di molti giochi di ruolo. Essi, pur concentrando l'attenzione dell'utente sullo sviluppo di abilità e caratteristiche del personaggio (*agent technology*) e talvolta appoggiandosi anche su una trama convincente, tralasciano molti aspetti fondamentali dell'*interactive storytelling*, come ad esempio le relazioni emozionali tra personaggi (l'interazione tra *characters* spesso si limita al combattimento o al baratto di oggetti).

Altre applicazioni ludiche che lambiscono il concetto di narrazione interattiva sono quelle definite *interactive fictions*: il personaggio controllato dall'utente deve muoversi attraverso lo *storyworld* cercando di raggiungere gli obiettivi che gli vengono assegnati. Solitamente la possibilità d'azione del giocatore si limita ad una scelta tra un elenco di azioni predefinite, talvolta applicabili ad oggetti che possono trovarsi all'interno dello *stage* (ad esempio *Raccogli + PozioneMagica*).

Ma a dispetto del nome, tali applicazioni non possono essere considerate interattive: solitamente, infatti, l'illusione di interattività è data da una strategia *foldback scheme*, la quale guiderà il personaggio sempre verso l'unico ed inevitabile finale del gioco. Inoltre, come già detto in merito ai giochi di ruolo, l'interazione tra i personaggi dello *storyworld* risulta spesso estremamente schematica e poco sofisticata.

Vi sono inoltre applicazioni pensate apposta per il Web, ovvero l'*hypertext fiction*, racconti testuali dinamici suddivisi in più pagine collegate attraverso link ipertestuali. I *digital storytelling softwares* invece sono poco più che semplici strumenti di raccolta ed assemblamento di immagini e video, privi di qualunque riferimento al concetto di interattività.